Transformation of Cognitive Maps

Yuan Miao 1, ChunYan Miao 2, XueHong Tao 1, ZhiQi Shen 2, and ZhiQiang Liu 3

1 Victoria University, PO Box 14428, Melbourne, VIC 8001, Australia
2 Nanyang Technological University, Nanyang Ave, Singapore 639798
3 City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

Abstract - Cognitive Maps (CMs), Fuzzy Cognitive Maps (FCMs) and Dynamical Cognitive Networks (DCNs) are related tools for modelling the cognition of human beings and facilitating machine inference accordingly. FCMs extend CMs; DCNs extend FCMs. Domain experts often face the challenge that CMs/FCMs are not sufficiently capable in many applications and DCNs are too complex. This paper presents a simplified DCN (sDCN) that extends the modelling capability of FCM/CM yet maintains simplicity. Additionally, this paper proves that there exists a theoretical equivalence among models in the cognitive map family of CMs, FCMs and sDCNs. It shows that every sDCN can be represented by an FCM or a CM, and vice versa; similarly, every FCM can be represented by a CM, and vice versa. The result shows that CMs, FCMs and sDCNs are a family of cognitive models that differs from many extended models. This paper also provides a constructive approach to transform one cognitive map model into other cognitive map models. Therefore, domain experts are able to model applications with more descriptive sDCNs and leave theoretical analysis to the simpler CM forms. The existence of theoretical transformation links among the models provides strong support for their theoretical analysis and flexibility in their applications.

Index Terms – Fuzzy Cognitive Map, Dynamical Cognitive Network, Equivalence, Transformation

I. INTRODUCTION

Cognitive Maps (CMs) [1], Fuzzy Cognitive Maps (FCMs) [6] and Dynamic Cognitive Networks (DCNs) [11] are related cognitive models. Cognitive Maps are the first model of the family and laid a foundation for other models. CMs have been applied in various application domains, for example analysis of electrical circuits [15], analysis and extension of graph-theoretic behavior [19], and plant control modelling [3]. CMs are easy to use and straightforward models; however, they do not differentiate the strength of relationships. Rather, each node simply makes its decision based on the number of positive impacts and the number of negative impacts: hence, a CM is an oversimplified model for many applications.

FCMs extended CMs by introducing fuzzified weights to describe the strength of the relationships. Fuzzified concepts and fuzzified causal relationship modelling - including FCMs - have gained comprehensive recognition and been widely applied in entertainment [14], games [2], multi-agent systems [7], social systems [3], ecosystems [4], financial systems [27], and earthquake risk / vulnerability analysis [28].

The construction of cognitive maps was mainly undertaken by domain experts in the early years. Recently, several data-oriented approaches - including Hebbian rules, the Swan Algorithm and differential techniques ([21],[22],[23],[24],[25],[26]) - have been proposed to train maps or networks for knowledge representation. Nevertheless, models from domain experts remain a key source for cognitive maps: they are especially suitable to the highly nonlinear, discrete causal linked maps, which are widespread in decision intensive causal systems.

In [11] it has been pointed out that FCMs are still not capable of handling complex causal systems: several core modelling capabilities are missing in the FCM model. FCMs do not differentiated the strength of factors - for example, a FCM may tell if a terrorism threat exists or not, but does not differentiate serious threats from minor threats. Additionally, FCMs do not describe the dynamics of the relationships. There is no differentiation between long-term threats and immediate threats.

To address the limitations of CMs and FCMs, several extensions have been proposed ([9],[10],[11],[12]). These proposals introduced more values to concepts including real valued concepts, nonlinear weight, and time delays. DCNs are the most powerful model in the family: they systematically address the modelling defects of other cognitive maps and are able to handle complex dynamic causal systems. On the other hand, DCNs are also very complex models. Their complexity has restricted their application, as domain knowledge experts are especially unfamiliar with dynamic DCN models. Many research ([17],[18],[15],[20]) and applications in recent years are still based on FCMs, because they are more capable than CMs but their analysis and applications are less complex than those of DCNs.

DCNs are more capable than either FCMs or CMs in modelling cognitive knowledge: nonetheless, CMs have a binary form which associates better with digital systems and logic foundations. Domain experts and researchers in the discipline would benefit greatly if the advantages of the models could be combined. To achieve that objective, this paper presents a simplified DCN, which is named sDCN, that balances the capability of DCNs and the simplicity of the widely familiar models CM and FCM models. For simplicity, sDCNs do not support infinite state sets or real intervals, nor model the continuous dynamics of how causal impact is built up. In contrast, sDCNs are able to model the strength of causes and impacts, which is one of the main modelling capabilities missing from
FCMs. (Dynamic relationships in sDCNs will be explored in a separate article balancing simplicity and generic DCNs.)

A significant contribution of the paper is that it reveals certain theoretical mutual modelling characteristics among the CM, FCM and sDCN cognitive models. Additionally, the paper provides a constructive approach to transform one cognitive model into other cognitive models: using this approach, domain experts could use one model to capture the causal knowledge and transform it into another model for theoretical analysis.

The rest of the paper is organized as follows: Section 2 compares three major different cognitive models, followed by the proposal of sDCNs in Section 3; Section 4 proves the theoretical equivalence between CMs, FCMs and sDCNs. Section 5 illustrates (by example) the analysis of sDCNs by comparing three major different cognitive models, followed by theoretical mutual modelling characteristics among the separate article balancing simplicity and generic DCNs.)

II. COGNITIVE MAPS

A The Cognitive Map - an over simplified model

Cognitive Maps (CMs) were used by Axelrod [1] for visualizing causal relationships among factors to facilitate human cognitive thinking. CMs use binary concepts to model important factors and binary links to model their causal relationships. A CM can be viewed as a tuple,

\[ M = \langle V, A \rangle, \]

where \( V \) is the set of vertices representing the concepts and \( A \) is the set of arcs representing the causal relationships among concepts.

\[ V = \{< v_1, f_{v_1}, S(v_1)>; ..., <v_n, f_{v_n}, S(v_n)>\}, \]

\[ A = \{<a(v_i, v_j), w(a(v_i, v_j))> | v_i, v_j \in V\}, \]

where \( v_i (i=1, 2, ..., n) \) are the vertices (concepts); \( n \) is the number of concepts; \( f_{v_i} \) is the decision function of \( v_i \); \( S(v_i) \) is the state set of \( v_i \); \( a(v_i, v_j) \) is the arc from \( v_i \) to \( v_j \); \( w(a(v_i, v_j)) \) is the weight of the arc \( a(v_i, v_j) \), which can also be written as \( w(v_i, v_j) \) or \( w_{ji} \). For simplicity, we do not differentiate \( v_i \in V \) from \( <v_i, f_{v_i}, S(v_i)> \in V \) when there is no ambiguity.

In CM, \( w_{ji} \in \{-1, 0, +1\}; -1 \) indicates a negative causal relationship; \( +1 \) represents a positive causal relationship; \( 0 \) means no causal relationship exists. This paper does not differentiate \( S(v_i) \) and \( S_v \), when no ambiguities is involved. For simplicity, we also do not differentiate \( v_i \in V \) from \( <v_i, f_{v_i}, S(v_i)> \in V \) when there is no ambiguity.

For each concept \( v_i, i=1, 2, ..., n, \) the decision making function \( f_{v_i} \) is defined as follows:

\[ f_{v_i}(u) = f_{v_i}\left(\sum_{j=1}^{n} w_{ij} x_j\right), \]

where \( x_j \) is the current state of vertex \( v_j \), and \( u \) is the total impact \( v_i \) received. Based on the causal inputs, the decision function decides the following state of the concept. CM uses binary concepts:

\[ S(v_i) = \{1, 1\}. \]

In some literature, \( \{0, 1\} \) is used instead.

As CMs do not recognize the different strengths of the causal relationships, they are not suitable to model complex systems. For instance, consider a decision-making process related to organizing business trips. A CM models the causal relationship of global/regional business demands (\( v_1 \)) and organizing business trips (\( v_3 \)) as

\[ V_1 + V_3 \]

Figure 2.1.1 CM of global/regional business demands (\( v_1 \)) and organizing business trips (\( v_3 \)).

That is to say, global/regional business demands have a positive impact on organizing business trips. When there are global/regional business demands, the decision to organize business trips would be made. The decision function of organizing business trips (\( v_3 \)) is

\[ f_{v_3}(u) = \begin{cases} 1, & u > 0 \\ 0, & u \leq 0 \end{cases}, \]

where \( u \) is the impact \( v_3 \) receives. It tells that positive impact leads to the decision of organizing business trips while negative impact leads to the opposite decision.

Similarly, CM can model the causal relationship between terrorism threats (\( v_2 \)) and organizing business trips (\( v_3 \)) as

\[ V_2 \rightarrow V_3 \]

Figure 2.1.2 CM of terrorism threats (\( v_2 \)) and organizing business trips (\( v_3 \)).

CM is a visualized, easy to use model for capturing human causal knowledge. Unfortunately, it creates difficulties in real applications due to its over-simplicity. When global/regional
Business demands ($v_1$) and terrorism threats ($v_2$) coexist, the cognitive model fails to facilitate the decision making.

For each concept $v_i$, $i=1, 2, ..., n$, the decision making function $f_{v_i}$ is defined as follows:

$$f_{v_i}(u) = f_{v_i} \left( \sum_{j=1}^{n} w_{ij} \times x_j \right)$$

where $x_j$ is the current state of vertex $v_j$. Based on the causal inputs, the decision function decides the next state of the concept. FCMs also use binary concepts:

$$S(v_j) = \{-1, 1\}$$

When FCM is applied in the decision-making process of organizing business trips, it uses weights to differentiate causal relationships. For example, the causal relationship between global/regional business demands ($v_1$) and organizing business trips ($v_3$) can be modeled as global/regional business demands have high positive impact on the decision of organizing business trips (Figure 2.2.1).

For example, the causal relationship between terrorism threats ($v_2$) and organizing business trips ($v_3$) can be modeled as terrorism threats ($v_2$) have very high negative impact on the decision of organizing business trips ($v_3$).

When global/regional business demands and terrorism threats coexist (Figure 2.2.3), the decision maker receives a total impact ($u$) of

$$u = 0.65 \times 1 + (-0.8) \times 1 = -0.15$$

The concept organizing business trips receives a total impact of zero.

### B The Fuzzy Cognitive Map - a model for simple causal links

FCMs are an extension of CMs created by introducing fuzzy weights to differentiate the strengths of causal relationships. An FCM can also be viewed as a tuple,

$$\mathbf{M} = \langle \mathbf{V}, \mathbf{A} \rangle$$

where $\mathbf{V}$ is the set of vertices representing the concepts and $\mathbf{A}$ is the set of arcs representing the causal relationships among concepts.

$$\mathbf{V} = \{ v_1, f_{v_1}, S(v_1) \}, \{ v_2, f_{v_2}, S(v_2) \}, \ldots, \{ v_n, f_{v_n}, S(v_n) \}$$

$$\mathbf{A} = \{ a(v_i, v_j) \mid w(a(v_i, v_j)) > 0 \land v_i, v_j \in \mathbf{V} \}$$

where $v_i$ ($i=1, 2, ..., n$) are the vertices (concepts); $n$ is the number of concepts; $f_{v_i}$ is the decision function of $v_i$; $S(v_i)$ is the state set of $v_i$; $a(v_i, v_j)$ is the arc from $v_i$ to $v_j$; $w(a(v_i, v_j))$ is the weight of arc $a(v_i, v_j)$, which can also be written as $w(v_i, v_j)$ or $w_{ji}$. In an FCM, $w_{ji}$ becomes a fuzzy description of the causal relationship. $w_{ji} < 0$ indicates a negative causal relationship, $w_{ji} > 0$ represents a positive causal relationship, $||w_{ji}||$ is the strength of the causal relationship.

When global/regional business demands and terrorism threats coexist (Figure 2.2.3), the decision maker receives a total impact ($u$) of

$$u = 0.65 \times 1 + (-0.8) \times 1 = -0.15$$

---

1 $\{0, 1\}$, or ternary value set $\{-1, 0, 1\}$ is adopted in some literatures.
Therefore, the decision towards organizing business trips is negative, because the decision function is

\[
\begin{align*}
    f_{v_i}(u) = \begin{cases} 
        1 & u > 0 \\ 
        0 & u \leq 0 
    \end{cases}.
\end{align*}
\]

This example shows that FCMs are more capable in cognitive modelling than CMs. Nevertheless, FCM moves just one step forward: if different factors connect to form a complex causal network, FCM may again fail to facilitate decision-making. Figure 2.2.3 shows that FCMs differentiate the strength of different links (different weights of the links), but fail to model the strength of the cause (no presentation of the different significances of the causes). Terrorist threats have been occurring frequently since 9/11, but people still travel: it does not mean that terrorism threats have a weaker causal link to the decision. Instead, people travel when terrorism threats are minor, therefore the impact is not as significant as the impact of global/regional business demands. FCMs’ inability to model the strength of the cause means they are unable to properly model the impact received by the decision function of organizing business trips. Additionally, FCM does not model how strong the decision of organizing business trips is, which affects the subsequent decisions. The inaccuracy could easily propagate through the causal network, which in the end leads to an unusable inference.

C. A General Dynamical Cognitive Network - a comprehensive but complex model

Dynamical Cognitive Networks (DCNs)\[11\] are general cognitive maps which model the strength of the cause, the impact and the causal relationship, and the dynamics of how the causal impact is built up. A DCN can also be viewed as a tuple,

\[
M = (V, A),
\]

where \(V\) is the set of vertices representing the concepts and \(A\) is the set of arcs representing the causal relationships among concepts.

\[
V = \{<v_1, f_{v_1}, S(v_1)>, <v_2, f_{v_2}, S(v_2)>, \ldots, <v_n, f_{v_n}, S(v_n)> \}, \quad (2.3.2)
\]

\[
A = \{<a(v_i, v_j), w(a(v_i, v_j))> | v_i, v_j \in V \}, \quad (2.3.3)
\]

where \(v_i (i = 1, 2, \ldots, n)\) are the vertices (concepts); \(n\) is the number of concepts; \(f_{v_i}\) is the decision function of \(v_i\); \(S(v_i)\) is the state set of \(v_i\), \(S(v_i)\) can be either a discrete set (with no restriction whether it has limited number of values), or a continuous interval; \(a(v_i, v_j)\) is the arc from \(v_i\) to \(v_j\); \(w(a(v_i, v_j))\) is the generic weight of arc \(a(v_i, v_j)\), which can also be written as \(w(v_i, v_j)\) or \(w_{ji}\). A generic weight can be a scalar weight, or a dynamic link. Several models can be applied in describing the dynamic link, one of which is the transfer function:

\[
y_{ij}(s) = w_{ij}(s) \times x_j(s), \quad (2.3.4)
\]

\[
x_j(t) = f_{v_j}(y_{i1}(t), y_{i2}(t), \ldots, y_{in}(t)), \quad (2.3.5)
\]

where \(y_{ij}(s)\) is the Laplace transform of \(y_{ij}(t)\), \(y_{ij}(t)\) is the impact from vertex \(j\) to vertex \(i\); \(t\) is the time, \(x_j(s)\) is the Laplace transform of \(x_j(t)\), \(w_{ij}(s)\) is the transfer function describing the dynamics of the impact; \(f_{v_j}\) is the decision function of \(v_j\). Decision functions in the DCN are no longer restricted as threshold functions. A detailed definition of the decision functions in DCNs can be found in [11].

DCNs avoid the structural non-robustness of CMs and FCMs, which makes them capable of modelling large complex causal systems \[11\]. Figure 2.3.1 shows a DCN model of the decision of organizing business trips.
where

\begin{align}
y_{31}(s) &= \frac{0.65}{s+1} \times x_1(s), \\
y_{32}(s) &= \frac{-0.8}{0.2s+1} \times x_2(s), \\
x_3(t) &= f_{y_3}(y_{31}(t), y_{32}(t)).
\end{align}

\begin{align}
y_{31}(s), y_{32}(s), x_1(s), x_2(s) \text{ are Laplace transformations of } y_3(t), \\
y_{32}(t), x_1(t), x_2(t), y_{31}(t) \text{ is the impact from } v_1 \text{ to } v_3, \\
y_{32}(t) \text{ is the impact from } v_2 \text{ to } v_3. 
\end{align}

As a general DCN allows continuous dynamics, the time is represented by \( t \) instead of step \( k \). A transfer function based on a Laplace transform is one of the most widely used approaches for dynamics. How to use transfer functions in Laplace transformations for modelling dynamic relationships can be found in [11].

The DCN model indicates that when global/regional business demands are strong (0.8) (bigger icon in the figure 2.3.1) but the impact builds up gradually (smooth transition curve); the terrorism threats are mild (0.5) (smaller icon in the figure 2.3.1), but the impact builds up quickly (steeper transition curve); the link of the terrorism threats is also stronger; the total final impact on the decision of organizing business trips is:

\[ +0.65 \times 0.8 + (-0.8) \times 0.5 = +0.24 \rightarrow 0.3. \]  

For simplicity, (2.3.9) omitted the transition but just gives the total final impact. The decision function starts with a negative decision and later turns into a positive decision, and the strength of the final decision is weak (0.3). When the terrorism threats become more significant, the decision may turn against traveling until the terrorism threats level drops.

A general DCN describes not only the strength of causes, impacts and effects, but also the dynamics of how impacts are built up. For example, the impact of global/regional business demands may take time to build up gradually, while a terrorism threat could have much more immediate effect. Such dynamics may affect the transition characteristics of the causal system, or even the final hidden pattern of the system, due to high nonlinearity. In the global/regional business demands example, the dynamics are modeled with transfer functions.

A general DCN is more complex than a CM or FCM. Although simplification of complex CMs and DCNs has been reported \([9]\), applying DCNs is still more difficult than applying CMs and FCMs. Additionally, experts and engineers who apply cognitive map are normally from disciplines in which state space equations, differential equations or transfer functions are not common tools. Regardless of the fact that DCNs allow scalar weight and binary concepts, the theory appears to be complex. All these issues have restricted the application of general DCNs; thus, development of a series of special DCNs for different needs is highly desirable.

This paper will present one simplified DCN (sDCN) which includes fuzzified concepts and fuzzified causal relationships: it is a model designed to be easily adopted by domain experts who are familiar with FCM and CMI, and is substantially extended to avoid the problems described in section 2.1 and 2.2. More importantly, this sDCN holds the important property that it can be transformed among CM and FCM models: therefore, it provides a tool that is easy for domain experts to use and guarantees the essential characteristics of CM and FCM models. This property is not present in many other CM and FCM extensions, including the generic DCNs. (The simplification of dynamic impacts without adopting advanced system tools like Laplace transformations will be reported separately.)

### III. A Simplified DCN

A significant contribution of FCMs/CMs is that they allow circles, which represent the feedback in a real system. Feedback / closed loops are widespread in real systems, including causal systems: however, FCMs use binary vertices to represent concepts. A binary concept has two states - either is or is not: in contrast, a closed loop represents the causal processes that need a few rounds to reach the final state. The mismatch between binary concepts and closed loops can lead to contradictory inference in FCMs. The following paragraph uses an example to illustrate binary concepts with iterated inference.

Suppose being rich is a binary concept: in other words, one is either rich or not rich. If a man has 1 dollar, he is not rich; if he is not rich, giving him 1 dollar does not make him rich. By iteration, after he receives dollars one by one until he owns multimillion dollars, he is still not rich. The contradiction is caused by the inability to differentiate different levels of being rich. If a fuzzy set is adopted, the membership of being rich can include fuzzy descriptions of not rich, a little rich, somewhat rich and rich: thus, thanks to the corresponding fuzzy logic, the contradiction can be avoided. This argument leads
to the adoption of a cognitive model which can be viewed as an sDCN.

An sDCN is defined as a tuple,

\[ M = \langle V, A \rangle, \]

where \( V \) is the set of vertices representing the concepts and \( A \) is the set of arcs representing the causal relationships among concepts.

\[ V = \{ v_i, \ f_i, S(v_i) > \}, \]

\[ A = \{ a(v_i, v_j), w(a(v_i, v_j)) \} \]

Here \( v_i \ (i=1, 2, \ldots, n) \) are the vertices (concepts); \( n \) is the number of concepts; \( f_i \) is the decision function of \( v_i \); \( S(v_i) \) is the finite state set of \( v_i \); \( a(v_i, v_j) \) is the arc from \( v_i \) to \( v_j \); \( w(a(v_i, v_j)) \) is the weight of arc \( a(v_i, v_j) \), which can also be written as \( w(v_i, v_j) \) or \( w_{ji} \). In sDCN, \( w_{ji} \) is the fuzzy description of the causal relationship. \( w_{ji} < 0 \) indicates a negative causal relationship, \( w_{ji} > 0 \) represents a positive causal relationship, \( \| w_{ji} \| \) is the strength of the causal relationship.

The decision making function of sDCN is also defined as \( f_{vi} \),

\[ f_{vi}(u) = f_{vi} \left( \sum_{j=1}^{n} w_{ij} \times x_j \right), \]

\[ u = \sum_{j=1}^{n} w_{ij} \times x_j. \]

Based on the causal inputs, the decision function decides the following state of the concept. The state spaces of sDCN concepts are finite value sets:

\[ S(v) = \{ x^1, x^2, \ldots, x^R \}, \]

\[ i = 1, 2, \ldots, n. \]

where \( n \) is the number of the concepts, \( R_i \) is the number of values the concept \( v_i \) has. In sDCN, each concept has its own value set with definitions depending on the needs of the system to be modeled. The state space of the sDCN is defined as the product of the vertices’ state space:

\[ S(M) = \prod_{i=1}^{n} S(v_i) \]

Now the sDCN is able to model cause and effect: it is better in logical inference. For example, if the terrorism threats are minor and the business need is important and urgent, the travel should be arranged; otherwise, the trip should be deferred to a later time. The sDCN is shown in Figure 3.1.
Theorem 4.1.1 The inclusive relationship $\subseteq$ of cognitive model instances is a transitive relationship.

Proof: Given three cognitive model instances, $M_1 \subseteq M_2$, $M_2 \subseteq M_3$.

As $M_1 \subseteq M_2$, there exist two constants $\alpha_1$ and $\beta_1$, for any value $x_{i1}^*$, of any concept $vi_1$ of $M_1$, there exists a value $x_{i2}^*$ of a concept $vi_2$ of $M_2$, that for all $x_{i2}(k) = x_{i2}^*$, $k \in \{1, 2, \ldots\}$, $\Rightarrow x_{i2}(\alpha_1 k + \beta_1) = x_{i2}^*$.

As $M_2 \subseteq M_3$, there exist two constants $\alpha_2$ and $\beta_2$, for any value $x_{i2}^*$ of any concept $vi_2$ of $M_2$, there exists a value $x_{i3}^*$ of a concept $vi_3$ of $M_3$, that for all $x_{i3}(k) = x_{i3}^*$, $k \in \{1, 2, \ldots\}$, $\Rightarrow x_{i3}(\alpha_2 k + \beta_2) = x_{i3}^*$.

That is, there exist two constants $\alpha_3$ and $\beta_3$, for any value $x_{i3}^*$, of any concept $vi_3$ of $M_3$, there exists a value $x_{i3}^*$ of a concept $vi_3$ of $M_3$, that for all $x_{i3}(k) = x_{i3}^*$, $k \in \{1, 2, \ldots\}$, $\Rightarrow x_{i3}(\alpha_3 k + \beta_3) = x_{i3}^*$, where $\alpha_1 = \alpha_2\alpha_3$, $\beta_1 = \alpha_2\beta_3 + \alpha_1\beta_2$. By definition, this proves that $M_1 \subseteq M_3$.

End Proof

Definition 4.1.2 Given two cognitive models $M$ and $M'$, $M$ is defined as being included in $M'$ if any instance of $M$, denoted as $M$, there exists an instance of $M'$, denoted as $M'$, such that $M \subseteq M'$.

The inclusive relationship of two cognitive models specifies that each instance of one model can be included in an instance of the other model. Obviously, $CM$ is included in FCM.

Theorem 4.1.2 Inclusive relationship $\subseteq$ of cognitive models is a transitive relationship.

Proof: Given three cognitive models, $M_1 \subseteq M_2$, $M_2 \subseteq M_3$.

As $M_1 \subseteq M_2$, for any $M_1 \in M_1$, there exists $M_2 \in M_2$,

such that $M_1 \subseteq M_2$.

As $M_2 \subseteq M_3$, there exists $M_3 \in M_3$, such that $M_2 \subseteq M_3$.

By Theorem 4.1.1, it holds $M_1 \subseteq M_3$.

which proves that $M_1 \subseteq M_3$.

End Proof

Definition 4.1.3 Two cognitive models, $M$ and $M'$, are equivalent iff $M \subseteq M'$ and $M' \subseteq M$.

The equivalent relationship of two cognitive models specifies that each instance of one model can be included in an instance of the other model.

If $M$ and $M'$ are equivalent, it is denoted as $M \equiv M'$.

Theorem 4.1.3 An equivalent relationship $\equiv$ of cognitive models is a symmetric relationship.

Proof: The proof can be derived directly from the Definition 4.1.3.

End Proof

4.2 The Equivalence between CMs, FCMs and sDCNs

Although CMs, FCMs and sDCNs are different cognitive models, this section proves that equivalence exists among them. It is straightforward, by the Definition 4.1.2, that CM $\subseteq$ FCM and FCM $\subseteq$ sDCN. The interesting part is that if sDCN is also included in CM, the equivalence among all the three models holds, by the transitive characteristics of the $\subseteq$ relationship.

Theorem 4.2.1 For any CM, $M$, there exists an FCM $M'$, $M \subseteq M'$ (CM $\subseteq$ FCM).

Proof: This proof will construct the $M'$ according to the given $M'$.

Let $M' = \varnothing$, // Start with an empty FCM model.

For each $v_i \in M$, $i = 1, 2, \ldots, n$, // $M$ has $n$ concepts.

add $v_i$ to $M'$:

$$M' = M' \cup \left\{ \left( v_i', f_{v_i'}, S(v_i') \right) \right\}.$$  \hspace{1cm} (4.2.1)

\footnote{Bold M is used to represent a cognitive model, normal M is used to denote an instance of the model. For example, CM is cognitive map model while CM is a cognitive map.

\footnote{if is an abbreviation meaning if and only if.}
\[ f_{v'_i \mid v'_j \in M'} = f_{v_i \mid v_j \in M}, \]
\[ S(v'_i) \mid v'_j \in M' = S(v_i) \mid v_j \in M \subset \{0, 1\}. \]
(4.2.2)

Here, for the simplicity of presentation, we do not differentiate \( v_i \in M \) and \( \bar{v_i} \in M' \) if no ambiguity is caused. Similarly, for simplicity of presentation, \( M' = M' \cup \left\{ \left\{ f_{v_i} \mid v'_j \in M' \right\}, S(v'_i) \mid v_j \in M \right\} \) is used here to represent \( V_{M'} = \left\{ v'_i \mid v_j \in M' \right\} \).

For each \( \alpha(v'_i, v'_j) \in M' \), \( \forall i, j = 1, 2, \ldots, n \), add \( \alpha(v'_i, v'_j) \) to \( M' \):
\[ M' = M' \cup \left\{ \left\{ \alpha(v'_i, v'_j), w(v'_i, v'_j) = w_{\bar{v}} \mid (v'_i, v'_j) \in M \right\} \right\}. \]
(4.2.3)

Set
\[ x_{v'_i} (0) \mid v'_j \in M' = x_{v_i} (0) \mid v_j \in M. \]
(4.2.4)

It can be verified that
\[ x_{v'_i} (k) \mid v'_j \in M' = x_{v_i} (k) \mid v_j \in M. \]
(4.2.5)

By definition, it proves that \( M \subset M' \), and \( CM \subset FCM \).

End Proof

Similarly, it holds that \( FCM \subset sDCN \).

Theorem 4.2.2 For any FCM, \( M \), there exists an sDCN \( M' \), \( M \subset M' \) (FCM \( \subset \) sDCN).

Proof:
This proof will construct the \( M' \) according to the given \( M \).

Let
\[ M' = \phi, \]
(4.2.1)
// Start with an empty sDCN model.

For each \( v_i \in M, i = 1, 2, \ldots, n \),
\( M \) has \( n \) concepts.
\[ M' = M' \cup \left\{ \left\{ v'_i \mid v'_j \in M' \right\}, f_{v'_i} \mid v_j \in M \right\}, \]
(4.2.6)
\[ f_{v'_i} \mid v'_j \in M' = f_{v_i} \mid v_j \in M, \]
(4.2.7)
\[ S(v'_i) \mid v'_j \in M' = S(v_i) \mid v_j \in M \subset \{0, 1\}. \]
(4.2.8)

For each \( \alpha(v'_i, v'_j) \in M' \), \( \forall i, j = 1, 2, \ldots, n \),
\( M' \)
\[ M' = M' \cup \left\{ \left\{ \alpha(v'_i, v'_j), w(v'_i, v'_j) = w_{\bar{v}} \mid (v'_i, v'_j) \in M \right\} \right\}. \]
(4.2.9)

Set
\[ x_{v'_i} (0) \mid v'_j \in M' = x_{v_i} (0) \mid v_j \in M. \]
(4.2.10)

It can be verified that
\[ x_{v'_i} (k) \mid v'_j \in M' = x_{v_i} (k) \mid v_j \in M. \]
(4.2.11)

By definition, it proves \( M \subset M' \), and \( FCM \subset sDCN \).

End Proof

Note that although the sDCN has been extended from a CM and achieved the same goals as extended models of FCMs (like DCNs), the equivalence property still holds. Theorem 4.2.3 will prove that \( sDCN \subset CM \). The loop is then closed for the models to be mutually transformed because an inclusive relationship is transitive.

Theorem 4.2.3 For any sDCN, \( M \), there exists a CM \( M' \), \( M \subset M' \).

Proof:
Let
\[ M' = \phi, \]
// Start with an empty CM model.

Firstly, \( \forall x_{v'_i} \in S(v_i) = \{ x^1_i, x^2_i, \ldots, x^{R_i}_j \}, v_i \in M, i = 1, 2, \ldots, n \)
// For each state value of a concept in \( M \), \( \forall \)
\( M \) has \( n \) concepts, \( S(v_i) \) has \( R_i \) values, and \( x_{v'_i} \) is a state value from the set.
\[ M' = M' \cup \left\{ \left\{ \bar{v'_i} \mid v'_j \in M' \right\}, f_{v'_i}, S(v'_i) \right\}. \]
(4.2.12)

// add concept \( \bar{v'_i} \) to \( M' \).

Define
\[ f_{\bar{v'_i}}(u) = \begin{cases} 1 & u \geq 1 \\ 0 & u < 1 \end{cases}, \]
(4.2.13)
where \( u \) is the sum of inputs of concept \( \bar{v'_i} \).
\( S(\overline{v}_{ji}) = \{0,1\} . \) \hspace{1cm} (4.2.14)

Secondly, \( \forall X_p \in S(M), X_p = \begin{bmatrix} x_i^n \\ x_2^n \\ \vdots \\ x_n^n \end{bmatrix} \) \hspace{1cm} (4.2.15)

// For each state of \( M, X_p \)

\[
M' = M' \cup \left\{ (\overline{v}_{2p}, f_{v_{2p}}, S(\overline{v}_{2p})) \right\}
\]

\[
\bigcup \left\{ (a(v_{11}^p, \overline{v}_{2p}), w(v_{11}^p, \overline{v}_{2p})),
\hspace{1cm} \nonumber \right. \\
\left. (a(v_{12}^p, \overline{v}_{2p}), w(v_{12}^p, \overline{v}_{2p})),
\hspace{1cm} \nonumber \right. \\
\ldots,
\left. (a(v_{in}^p, \overline{v}_{2p}), w(v_{in}^p, \overline{v}_{2p})) \right\} \hspace{1cm} (4.2.16)
\]

// add concept \( \overline{v}_{3q} \) to \( M' \);
// add arcs from \( \overline{v}_{3q} \) to \( \overline{v}_{11}, \overline{v}_{12}, \ldots, \overline{v}_{in} \).

Set
\[ w(a(\overline{v}_{3q}, \overline{v}_{ji})) = +1 . \] \hspace{1cm} (4.2.20)

// Set the arc as a positive impact link.

where \( n \) is the sum of inputs of concept \( \overline{v}_{3q} \);
\[ S(\overline{v}_{3q}) = \{0,1\} . \] \hspace{1cm} (4.2.22)

Thirdly, \( \forall X_q \in S(D(M)), X_q = \begin{bmatrix} x_i^q \\ x_2^q \\ \vdots \\ x_n^q \end{bmatrix} \) \hspace{1cm} (4.2.19)

// For each derived state of \( M, X_q \) (a derived state is a state that can be the resultant/derived state inferred from an initial state

\[
M' = M' \cup \left\{ (\overline{v}_{3q}, f_{v_{3q}}, S(\overline{v}_{3q})) \right\}
\]

\[
\bigcup \left\{ (a(\overline{v}_{11}^q, \overline{v}_{3q}), w(\overline{v}_{11}^q, \overline{v}_{3q})),
\hspace{1cm} \nonumber \right. \\
\left. (a(\overline{v}_{12}^q, \overline{v}_{3q}), w(\overline{v}_{12}^q, \overline{v}_{3q})),
\hspace{1cm} \nonumber \right. \\
\ldots,
\left. (a(\overline{v}_{in}^q, \overline{v}_{3q}), w(\overline{v}_{in}^q, \overline{v}_{3q})) \right\} \hspace{1cm} (4.2.24)
\]

\[ \text{where } F_M = \begin{bmatrix} f_1 \\ f_2 \\ \ldots \\ f_n \end{bmatrix} \text{ is the decision function vector of model } M. \]

// For all \( x_q, x_p \) such that \( F_M(X_p) = X_q \),
// add arc \( a(\overline{v}_{3q}, \overline{v}_{3q}) \) to \( M' \).

Set
\[ w(a(\overline{v}_{3q}, \overline{v}_{ji})) = +1 . \] \hspace{1cm} (4.2.25)

// Set the arc as a positive impact link.

Suppose \( X_{y(0)} = \begin{bmatrix} x_i^y \\ x_2^y \\ \vdots \\ x_n^y \end{bmatrix} \)

define \( V_i = \{ \overline{v}_{11}, \overline{v}_{12}, \ldots, \overline{v}_{in} \} \subset M' \).

For simplicity of presentation, if no ambiguity is caused, we use \( \{ \overline{v}_{11}, \overline{v}_{12}, \ldots, \overline{v}_{in} \} \subset M' \) to represent
Theorem 4.2.4

Denote the FCM formation among sDCN and FCM or CM can be performed by studying the construction of the CM that represents the FCM of a model. Given an instance inclusive model can be constructed. It provides an approach from which, given an instance inclusive model is proven. The proof in the paper is constructive. Therefore, the existence of an inclusive model can be more than one root between zero. But this proof does not tell what the root is. And there can be more than one root between x1 and x2; The second type of proof is a constructive approach, which is an approach to construct an inclusive model. Therefore, the existence of an inclusive model is proven. The proof in the paper is constructive. It provides an approach from which, given an instance map of one model, the corresponding inclusive model can be constructed.

For the clarity of presentation, this section provides a case study of constructing the CM that represents the FCM of the global/regional business demands in Figure 2.2.3. The transformation among sDCN and FCM or CM can be performed similarly according to the result in section IV.

V. APPLICATIONS OF SDCN AND THE TRANSFORMATION

A Transforming FCM to a representative CM

To prove the existence of an inclusive model, there are fundamentally two types of proofs: One type of proof is a proof which shows that such a model exists. By analogy, given a f(x) continuous; if f(x1)>0 and f(x2)<0, then there exists a root between x1 and x2 because the continuous f(x) has to pass zero. But this proof does not tell what the root is. And there can be more than one root between x1 and x2; The second type of proof is a constructive approach, which is an approach to construct an inclusive model. Therefore, the existence of an inclusive model is proven. The proof in the paper is constructive. It provides an approach from which, given an instance map of one model, the corresponding inclusive model can be constructed.

For the clarity of presentation, this section provides a case study of constructing the CM that represents the FCM of the global/regional business demands in Figure 2.2.3. The transformation among sDCN and FCM or CM can be performed similarly according to the result in section IV.

Denote the FCM M as

\[
\left\{ \begin{aligned}
&\left( \bar{v}_{11}, f_{v_{11}}, S(\bar{v}_{11}) \right), \\
&\left( \bar{v}_{12}, f_{v_{12}}, S(\bar{v}_{12}) \right), \\
&\ldots, \\
&\left( \bar{v}_{1n}, f_{v_{1n}}, S(\bar{v}_{1n}) \right) \right\} \subseteq M'.
\]

For all \( v \in V_I \), \( x_i(0) = 1 \);
For all \( v \not\in V_I \), \( x_i(0) = 0 \);

It can be verified that for any value \( x_i^{\gamma_i} \) of any concept \( v_i \),

\[
x_i | v_i \in M \ (k) = x_i^{\gamma_i},
\]

(4.2.26)

it holds that

\[
x_i^{\gamma_i} | v_i \in M' \ (2k) = 1.
\]

(4.2.27)

By Definition 4.1.1, \( M \subseteq M' \).

End Proof

Theorem 4.2.4 CM = FCM = sDCN.

Proof: The proof can be derived from the definitions, so has been omitted.

\[
V(M) = \{ v_1, v_2, v_3 \}, \quad \text{// simplified presentation (5.1)}
\]

\[
A(M) = \{ <a(v_1, v_3), w(v_1, v_3) = w_3 = 0.65 >, \\
< a(v_2, v_3), w(v_2, v_3) = w_{32} = -0.8 > \},
\]

(5.2)

\[
f_{v_{13}}(u) = \begin{cases} 
1 & u = w_{13} x_1 + w_{12} x_2 > 0 \\
0 & u = w_{13} x_1 + w_{12} x_2 < 0
\end{cases},
\]

(5.3)

Note that \( f_{v_1}, f_{v_2} \) are not used in this case study and have been omitted. The corresponding CM can be constructed as

\[
V(M') = \{ v_{11}, v_{12}, v_{13}, v_{21}, v_{22}, v_{23}, v_{24},
\]

\[
v_{13}^1 \text{ or } v_{13}^2 \text{ or } v_{13}^3 \}
\]

A technical simplicity has been performed to simplify the presentation, where one node can represent both \( v_{13}^1 \) and \( v_{13}^3 \) (as used in the Theorem 4.2.3). Similarly, one node can represent both \( v_{13}^2 \) and \( v_{13}^3 \)

\[
x_{11} | M' = 1 \text{ if } x_{11} | M = 0 , x_{11} | M' = 0 \text{ if } x_{11} | M = 1; \]

(5.4)

\[
x_{11} | M' = 1 \text{ if } x_{11} | M = 1 , x_{11} | M' = 0 \text{ if } x_{11} | M = 0; \]

(5.5)

\[
x_{12} | M' = 1 \text{ if } x_{12} | M = 0 , x_{12} | M' = 0 \text{ if } x_{12} | M = 1; \]

(5.6)

\[
x_{12} | M' = 1 \text{ if } x_{12} | M = 1 , x_{12} | M' = 0 \text{ if } x_{12} | M = 0; \]

(5.7)

\[
A(M') = \{ <a(v_{11}, v_3), w(v_{11}, v_3) = 1 >,
\]

\[
< a(v_{11}, v_2), w(v_{11}, v_2) = 1 >,
\]

\[
< a(v_{11}, v_3), w(v_{11}, v_3) = 1 >,
\]

\[
< a(v_{11}, v_2), w(v_{11}, v_2) = 1 >,
\]

\[
< a(v_{12}, v_1), w(v_{12}, v_1) = 1 >,
\]

\[
< a(v_{12}, v_2), w(v_{12}, v_2) = 1 >,
\]

\[
< a(v_{12}, v_3), w(v_{12}, v_3) = 1 >,
\]

\[
< a(v_{12}, v_2), w(v_{12}, v_2) = 1 >,
\]

\[
< a(v_{12}, v_3), w(v_{12}, v_3) = 1 >,
\]

\[
< a(v_{12}, v_2), w(v_{12}, v_2) = 1 >,
\]

\[
< a(v_{23}, v_{13}), w(v_{23}, v_{13}) = 1 >,
\]

\[
< a(v_{23}, v_{13}), w(v_{23}, v_{13}) = 1 >,
\]

\[
< a(v_{23}, v_{13}), w(v_{23}, v_{13}) = 1 >,
\]

\[
< a(v_{23}, v_{13}), w(v_{23}, v_{13}) = 1 >,
\]

(5.8)

\[
f_{v_{13}}(u) = \begin{cases} 
1 & u \geq 2 \\
0 & u < 2
\end{cases},
\]

(5.9)

\[
f_{v_{23}}(u) = \begin{cases} 
1 & u \geq 2 \\
0 & u < 2
\end{cases},
\]

(5.10)

\[
f_{v_{23}}(u) = \begin{cases} 
1 & u \geq 2 \\
0 & u < 2
\end{cases},
\]

(5.11)

\[
f_{v_{23}}(u) = \begin{cases} 
1 & u \geq 2 \\
0 & u < 2
\end{cases}.
\]

(5.12)
\[ f_{V_{ij}}(u) = \begin{cases} 1 & u \geq 1 \\ 0 & u < 1 \end{cases}, \quad (5.13) \]
\[ f_{V_{ij}}(u) = \begin{cases} 1 & u \geq 1 \\ 0 & u < 1 \end{cases}, \quad (5.14) \]

Note that CM is binary: therefore, the “\( f = 1 \) if \( u \geq 2 \)” means that both of the causes should be present; “\( f = 1 \) if \( u \geq 1 \)” means that either of the causes should be present. It can be verified that for any \( k = 1, 2, 3, \ldots \),

if \( x_i(k) = 0 \), then \( x_{ij}(2k) = 1 \), else \( x_{ij}(2k) = 0 \); \( i = 1, 2, 3 \)
if \( x_i(k) = 1 \), then \( x_{ij}(2k) = 1 \), else \( x_{ij}(2k) = 0 \); \( i = 1, 2, 3 \)

Therefore, the constructed CM can represent the FCM. Figure 5.1.1 illustrates the constructed CM.

**Figure 5.1.1 Constructed CM representing the FCM in Figure 2.2.3**

**B Analyzing the sDCN via the constructed CM**

Different models have different merits. sDCNs are more descriptive and allow domain experts to model complex systems easily; CMs have simpler structure and are more appropriate for theoretical analysis. The equivalence among the models provides the freedom for selecting the most suitable models for a given task. Via the constructed CM, this section proves that the final hidden patterns of sDCNs are either a static state or a limit cycle.

**Theorem 5.2.1** For any sDCN, \( M \), with any initial state \( x_0 \), the hidden pattern (final inference pattern) is either a static state, or a limit cycle.

**Proof:**

As sDCN = CM, there exists a CM, \( M' \), such that

\[ M \subseteq M'. \]

As \( M' \) is a CM, the final hidden pattern of \( M' \) is either a static state or a limit cycle. Therefore, the final hidden pattern of \( M \) is either a static state or a limit cycle.

**End Proof**

Similarly, further research into sDCNs or FCMs can be carried out by studying the inclusive models of other types, e.g. the inclusive CM. For example, [29] has several theoretical analysis results on CM, including the length of the result limit cycle; the corresponding analysis on sDCN would be rather complicated. This paper shows that the analysis can be performed through the inclusive CM.

**C Possible Hardware Implementations of Cognitive Maps**

To date, there has not been significant implementation of cognitive map tools or platforms. This paper shows that the construction of a CM corresponding to an FCM and sDCN is possible. The result proves that the implementation of cognitive map platforms can be based solely on CMs. CMs have binary concept values and binary relationship values; such a simple structure can significantly reduce the implementation cost and make hardware implementations possible. It is similar to computers, which have relatively simple instruction sets (assembly language level) with the CPU, but is able to support complex higher level languages, from Java, C# to 4GL.

VI. CONCLUSIONS AND DISCUSSIONS

Feedback loops are common characteristics of real causal systems. Both CMs and FCMs try to model feedback, but feedback needs a mechanism to indicate the strength of impacts. The binary concepts of CMs and FCMs fail to meet this requirement - a CM or FCM with circles can produce contradictory inference. To address the problem, several extensions of CMs and FCMs have been made, including DCNs. However, DCNs are too complex for many domain experts.

This paper proposes a simplified DCN model. sDCNs achieve several goals that motivated the earlier extensions to CMs and FCMs. The sDCNs are able to model the strength of the cause, the strength of the causal relationships and the degree of the impacts. Node state sets of sDCN are limited as finite state value sets. Real intervals are no longer used in sDCN. Even with this constraint, sDCNs are normally enough for domain experts to model their cognitive knowledge, and at the same time avoid unnecessary complexities. This paper proves that sDCNs, CMs and FCMs can be grouped as a family of cognitive models by the inclusive relationship discovered in this paper, while other cognitive map extensions are not necessarily having this property.

This paper defines an inclusive relationship. Model instance \( M_1 \) is said to be included in model instance \( M_2 \) if all inference patterns of \( M_1 \) are represented in the inference patterns of \( M_2 \).

If every model instance of a cognitive model (e.g. \( M_a \)) can find an inclusive model instance of another cognitive model (e.g. \( M_b \)), then \( M_a \) is said to be included in \( M_b \). The inclusive model instance is called a transformation of the original
model instance in Mb. This paper proves that CM, FCM and the sDCN are mutually inclusive. The proof of the corresponding results is constructive. For any given model instance in the family, it provides an approach to construct the corresponding inclusive model.

This result implies that domain experts can model applications with more descriptive sDCNs and leave theoretical analysis to the simpler CM forms. In Section V an example is given to prove the possible final patterns of sDCNs via the transformation. The length of the cycles in the final inference pattern can also be analyzed through an inclusive CM. Given a model instance, its inclusive model instances are not unique.

The transformation results may also open a new area of possible hardware implementation of cognitive maps. The fundamental structure of the hardware can be the simplest CM, with binary nodes and binary links. In theory, all FCM and sDCN model instances can find an inclusive CM model.

References


Yuan Miao and the other authors may include biographies at the end of regular papers. Biographies are not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author’s educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state or country, and year. The author’s major field of study should be in lowercase.

The second paragraph uses the pronoun of the person (he or she) and not the author’s last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author’s title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of
the biography. The final printed size of an author photograph is 2.54 cm wide by 3.18 cm long. Personal hobbies will be deleted from the biography.